**WEEK 5 LABORATORY ACTIVITY: RESTAURANT INVENTORY MANAGEMENT SYSTEM**

**Objective:** Design a database, write SQL queries, and create a basic React UI for a restaurant inventory system.

**Part 1: Database Design and SQL Queries**

**Instructions:**

1. **Study Tables & Entities:** Review the provided table names and their columns (see previous detailed instruction).
2. **CREATE TABLE Queries:** Write SQL CREATE TABLE statements for Category, Ingredients, StockIngredients, MenuItems, Recipes, and Orders tables, defining primary and foreign keys.

**Category Table:**

* + CategoryID (Integer, Primary Key)
  + Category (String)

**Ingredients Table:**

* + IngredientsID (Integer, Primary Key)
  + IngredientName (String, Not Null, Unique)
  + UnitOfMeasurement (String)
  + CategoryID (Integer, Foreign Key referencing Category.CategoryID)

**StockIngredients Table:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **StockIngredientsID** | **IngredientsID** | **Container** | **Quantity** | **Container\_Size** | **Container\_Price** | **Total\_Quantity/Stock** | **Total\_Price** | **Unit\_Price** |
| 1 | 1 | bottle | 5 | 1000 | 150 | 5000 | 750 | 0.15 |
| 1 | 1 | bottle | 10 | 750 | 120 | 7500 | 1200 | 0.16 |
| 1 | 1 | sachet | 10 | 100 | 25 | 1000 | 250 | 0.25 |

**MenuItems Table:**

* + MenuID (Integer, Primary Key)
  + Menu (String, Not Null, Unique)
  + SellingPrice (Decimal)

**Recipes Table:**

* + RecipeID (Integer, Primary Key, Auto Increment)
  + MenuID (Integer, Foreign Key referencing MenuItems.MenuID)
  + IngredientsID (Integer, Foreign Key referencing Ingredients.IngredientsID)
  + Quantity (Decimal)

**Orders Table:**

* + OrderID (Integer, Primary Key, Auto Increment)
  + MenuID (Integer, Foreign Key referencing MenuItems.MenuID)
  + Quantity (Integer)
  + TotalPrice (Decimal)
  + OrderDate (Timestamp, Default: Current Timestamp)

1. **CREATE VIEW Queries:** Write SQL CREATE VIEW statements for:
   * View\_TotalStockByIngredient (total stock per ingredient).
   * View\_TotalIngredientsUsed (total ingredient usage from orders).
   * View\_RemainingIngredients (remaining stock based on the above views).

**Part 2: Front-End UI Design using React JS (Group)**

**Instructions:**

**1. Data Display UIs (React):**

* **UI for Tables & Views:** Design user interfaces in React JS to display the data from all six database tables and the three created views.
* **Display Tables & Views:** Create separate React components (or a single component with tabs/sections) to display the data from each table and each view in a tabular format. Include relevant columns for each.
* **UI with Navigation:** Design a navigation bar to allow users to easily access the data display for all six database tables and the three created views.

**2. Data Insertion UIs (React):**

* **UI for Data Entry Forms:** Design user interfaces in React JS that include forms to insert new data into each of the six database tables.
* **Form Design:** Create separate React components (or integrate forms within the table display components) for each table. Each form should include input fields for all necessary columns of the respective table.
* **User-Friendly Forms:** Ensure the forms are user-friendly with appropriate labels, input types, and clear submission mechanisms.